Αναγνώριση Προτύπων

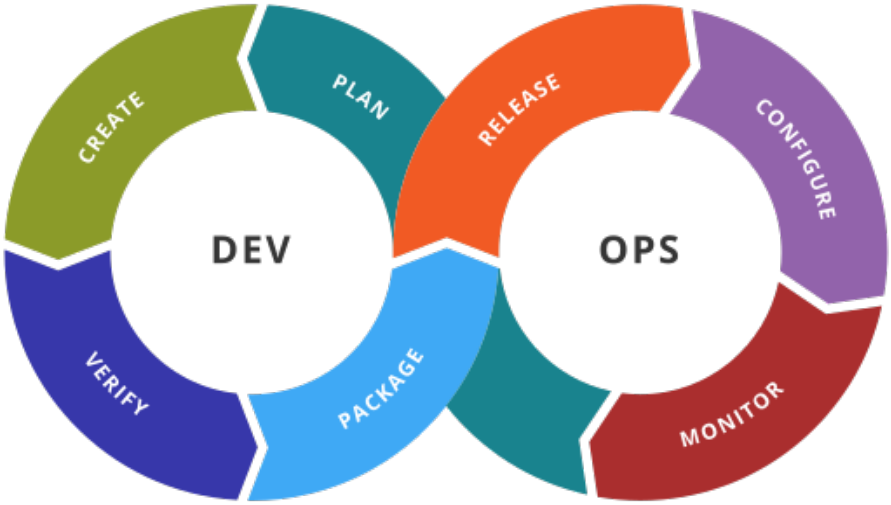
Εργασία μαθήματος – Περιγραφή Προβλήματος και Σετ Δεδομένων

1. Εισαγωγή

1.1. Ανάπτυξη Λογισμικού - DevOps Approach

Ο τρόπος ανάπτυξης λογισμικού μετασχηματίζεται, από τις παραδοσιακές τεχνικές ανάπτυξης λογισμικού (π.χ. μοντέλο καταρράκτη (waterfall)) προς νέες μεθοδολογίες και προσεγγίσεις. Μια ολοένα και πιο χρησιμοποιούμενη προσέγγιση αποτελεί η *Agile* μεθοδολογία, η οποία απαρτίζεται από ένα σύνολο αρχών για την ανάπτυξη λογισμικού, του οποίου οι απαιτήσεις αλλάζουν συχνά. Οι αλλαγές αυτές έχουν άμεση επίδραση τόσο στις λειτουργικές όσο και στις μη λειτουργικές απαιτήσεις του προϊόντος λογισμικού και η Agile προσέγγιση φροντίζει ώστε αυτές οι αλλαγές να γίνονται αποδεκτές και να προσαρμόζονται οι λύσεις που σχεδιάζονται, κυρίως μέσω της στενής συνεργασίας μεταξύ διεπιστημονικών, αυτοργανούμενων ομάδων.

Προς την Agile κατεύθυνση αναπτύχθηκε η DevOps προσέγγιση, η οποία αποτελεί μια «κουλτούρα» τεχνολογίας λογισμικού με στόχο την πρακτική ενοποίηση δύο βασικών πυλώνων: του **Software Development (Dev)** και του **Software Operations (Ops)**.



Εικόνα 1 – DevOps Toolchain

1.2. GitHub

Το GitHub είναι μια διαδικτυακή πλατφόρμα ανάρτησης, διατήρησης και ανάπτυξης λογισμικού η οποία, δημιουργήθηκε το 2008 από τους T. Preston-Werner, C. Wanstrath και P. J. Hyett και τα τελευταία χρόνια έχει γίνει εξαιρετικά δημοφιλής. Μέσω της πλατφόρμας δίνεται η δυνατότητα για ταυτόχρονη συνεισφορά πολλών developers σε ένα ή περισσότερα project, γεγονός που διευκολύνει τη διαδικασία ανάπτυξης λογισμικού. Μέσω της δραστηριότητας που καταγράφεται στο GitHub, γίνεται δυνατή η συλλογή πλήθους δεδομένων που μπορούν να χρησιμοποιηθούν στο κομμάτι της εξόρυξης γνώσης με απώτερο σκοπό την επίλυση προβλημάτων που άπτονται στον τομέα της τεχνολογίας λογισμικού. Ένα από τα προβλήματα αυτά

αποτελεί και η αναγνώριση προφίλ μηχανικών λογισμικού (software engineers profiles). Τα δεδομένα που μπορούν να χρησιμοποιηθούν προς την κατεύθυνση αυτή ανήκουν στις παρακάτω κατηγορίες:

- ✓ **Commits data**

Τα δεδομένα της κατηγορίας αυτής αποτελούν την καταγραφή όλων των ενεργειών που γίνονται από τους contributors και έχουν ως στόχο τη δημιουργία περιεχομένου στο αποθετήριο (ανέβασμα κώδικα, libraries, documentation κ.α.).

- ✓ **Issues data**

Τα δεδομένα της κατηγορίας αυτής αποτελούν την καταγραφή όλων των ενεργειών που γίνονται από τους contributors και έχουν ως στόχο την παρακολούθηση του προϊόντος λογισμικού με στόχο την εύρεση σφαλμάτων, καταγραφή και εμπλουτισμός λειτουργικών απαιτήσεων κ.α.

- ✓ **Repositories data**

Τα δεδομένα της κατηγορίας αυτής αποτελούν στατιστικά στοιχεία των αποθετηρίων που περιλαμβάνουν δεδομένα για το μέγεθος (number of commits, number of issues, number of files κ.α.), τους contributors (owner, number of contributors κ.α.)

Όλα τα παραπάνω δεδομένα είναι διαθέσιμα μέσω μιας καλά ορισμένης προγραμματιστικής διεπαφής (RESTful API) [1].

2. Εργασία

2.1. Το Πρόβλημα

Η παρούσα εργασία αναφέρεται στην αναγνώριση των ρόλων που αναλαμβάνουν οι μηχανικοί λογισμικού που συμμετέχουν σε έργα λογισμικού (στο GitHub). Όσον αφορά την επιλογή των repositories (συγκεκριμένα αποθετήρια λογισμικού), αυτή έγινε με βάση τη δημοφιλία τους, όπως αυτή αποτυπώνεται στον αριθμό των stars, καθώς και από το μέγεθός τους. Όσον αφορά το μέγεθος, τα αποθετήρια επιλέχθηκαν με στόχο να αποτελούν ένα τυπικό παράδειγμα έργου λογισμικού στο οποίο μπορεί να εφαρμοστεί η agile προσέγγιση. Χαρακτηριστικό παράδειγμα των projects αυτών αποτελεί μια ομάδα ανάπτυξης λογισμικού που αποτελείται από 5 – 10 άτομα (basic contributors).

Πιο συγκεκριμένα, το βασικό ζητούμενο του προβλήματος προς επίλυση αποτελεί η κατασκευή ενός συστήματος αναγνώρισης των ρόλων που αναλαμβάνουν οι μηχανικοί λογισμικού βασισμένοι σε μια σειρά από χαρακτηριστικά τα οποία μετρήθηκαν από τη δραστηριότητά τους στο GitHub.

- ➢ **Επιλογή των ομάδων υπό εξέταση**

Η διαδικασία αυτή περιλαμβάνει την επιλογή των ομάδων, οι οποίες ιδανικά θα προκύψουν από την ανάλυση των δεδομένων. Χαρακτηριστικά παραδείγματα ομάδων αποτελούν οι ομάδες **dev, ops** και **devops.** Η ομάδα dev αναφέρεται στους «καθαρούς» developers, η ομάδα ops στα operations και η ομάδα devops σε μηχανικούς, οι οποίοι φαίνεται να επιτελούν και τους δύο ρόλους. Μια ακόμα ομάδα που θα μπορούσε να αναλυθεί είναι οι μηχανικοί που αποτελούν εξωκείμενες τιμές (outliers) για μια σειρά από χαρακτηριστικά.

- **Εύρεση χαρακτηριστικών που δίνουν διακριτική ικανότητα μεταξύ dev και ops**

Η διαδικασία αυτή περιλαμβάνει μια διερεύνηση αναφορικά με τον τρόπο που μπορεί να αξιοποιηθεί η πληροφορία που υπάρχει στα χαρακτηριστικά που δίνονται με στόχο την εξαγωγή συμπερασμάτων αναφορικά με το ρόλο του κάθε μηχανικού. Μια ενδεικτική προσέγγιση αποτελεί η διενέργεια clustering και στη συνέχεια η επόπτευση των χαρακτηριστικών των μηχανικών που εντάσσονται σε κάθε cluster με στόχο τη διερεύνηση για το αν αυτό μπορεί να χαρακτηριστεί ως αντιπροσωπευτικό κάποιου ρόλου.

Πίνακας 1 Παράδειγμα εύρεσης προφίλ μηχανικών με βάση συγκεκριμένα χαρακτηριστικά [2]

Clusters	**Feature**					**Profile**
	Issues Participated	**Comments Made**	**Issues Opened**	**Issues Closed**	**Commits Authored**	
#1	Low	Low	Low	Low	High	**Pure Dev**
#2	High	High	High	Low	High	**DevOps**
#3	High	High	High	High	Low	**Project Owner**

Στον παραπάνω πίνακα φαίνεται το αποτέλεσμα της διενέργεια clustering με τη χρήση 6 χαρακτηριστικών. Από τα clusters που σχηματίστηκαν και από την εξέταση των χαρακτηριστικών των στοιχείων που ανήκουν σε καθένα από αυτά ταυτοποιήθηκε το προφίλ μηχανικού, το οποίο αντιπροσωπεύουν. Όπως φαίνεται από τον πίνακα, στο πρώτο cluster έχουμε τους μηχανικούς, οι οποίοι εμφανίζουν μεγάλη συνεισφορά μόνο στο κομμάτι των commits. Κατά συνέπεια χαρακτηρίζονται ως **pure devs**. Από την άλλη πλευρά, μηχανικοί με πολύ μικρή συνεισφορά στον κώδικα (commits), αλλά μεγάλη συνεισφορά στα σχόλια (comments) και τα issues, χαρακτηρίζονται ως **project owners (Ops)**.

2.2.Περιγραφή σετ δεδομένων

Τα δεδομένα τα οποία σας δίνονται περιλαμβάνουν τα εξής χαρακτηριστικά:

- **dataset.csv**

Στο αρχείο αυτό περιέχονται πληροφορίες αναφορικά όλους τους μηχανικούς που έχουν συνεισφέρει στα 240 most-stared GitHub projects, τα οποία έχουν 5 – 10 basic contributors. Τα χαρακτηριστικά με τα οποία θα χρησιμοποιηθούν για την αναγνώριση των ρόλων περιγράφονται στον παρακάτω πίνακα:

Μετρική	Περιγραφή
Contributor_login	Το username του contributor
repository_name	Το όνομα του αποθετηρίου με τη μορφή: **Owner_Name/Repository_Name**
total_contributions	Ο συνολικός αριθμός των commits που έχει κάνει
average_issues_comments_length	Το μέσο μήκος των σχολίων που έχει κάνει σε όλα τα issues που έχει κάνει σχόλια
average_time_to_close_issues	Ο μέσος χρόνος **σε ώρες** που χρειάζεται ο contributor για να κλείσει ένα issue.[1]
issues_closed	Ο αριθμός των issues έχουν κλείσει από τον contributor
issues_opened	Ο αριθμός των issues έχουν ανοίξει από τον contributor
issues_participated	Ο αριθμός των issues στα οποία ο contributor έχει συμμετάσχει
issues_closed_per_day	Ο μέσο αριθμός των issues έχουν κλείσει από τον contributor ανά μέρα
average_comments_per_issue	Ο μέσο αριθμός των comments που έχουν κλείσει από τον contributor ανά issue
early_additions	Ο αριθμός των γραμμών που έχουν προστεθεί από τον contributor στο πρώτο 20% του χρόνου δραστηριότητάς του
early_deletions	Ο αριθμός των γραμμών που έχουν αφαιρεθεί από τον contributor στο πρώτο 20% του χρόνου δραστηριότητάς του
late_additions	Ο αριθμός των γραμμών που έχουν προστεθεί από τον contributor στο τελευταίο 20% του χρόνου δραστηριότητάς του
late_deletions	Ο αριθμός των γραμμών που έχουν αφαιρεθεί από τον contributor στο τελευταίο 20% του χρόνου δραστηριότητάς του
change_bursts	Ο αριθμός των εξάρσεων δραστηριότητας [1]
biggest_burst_length	Το μήκος σε μέρες της μεγαλύτερης έξαρσης δραστηριότητας

[1] Ο χρόνος αυτός υπολογίζεται από το χρόνο που έχει παρέλθει από τη στιγμή που ανοίχθηκαν όλα τα issues που έχει κλείσει ο συγκεκριμένος contributor.

inactive_period_within_active_period	Το ποσοστό της περιόδου δραστηριότητας όπου ο contributor δεν ήταν ενεργός[2]
tot_additions	Ο συνολικός αριθμός των γραμμών που έχουν προστεθεί από τον contributor στο σύνολο του χρόνου δραστηριότητάς του
tot_deletions	Ο συνολικός αριθμός των γραμμών που έχουν αφαιρεθεί από τον contributor στο σύνολο του χρόνου δραστηριότητάς του
activity_period_in_days	Η περίοδος δραστηριότητας σε μέρες
total_file_additions	Ο συνολικός αριθμός των αρχείων που έχουν προστεθεί από τον contributor στο σύνολο του χρόνου δραστηριότητάς του
total_file_deletions	Ο συνολικός αριθμός των αρχείων που έχουν αφαιρεθεί από τον contributor στο σύνολο του χρόνου δραστηριότητάς του
total_file_modifications	Ο συνολικός αριθμός των αρχείων που έχουν μεταβληθεί από τον contributor στο σύνολο του χρόνου δραστηριότητάς του
total_file_changes	Ο συνολικός αριθμός των αρχείων που έχουν γίνει αλλαγές
commits_authored	Ο συνολικός αριθμός των commits που έχει κάνει
total_lines_of_code_changed	Ο συνολικός αριθμός των γραμμών κώδικα που έχει αλλάξει ο contributor
violations_added	Ο συνολικός αριθμός violations που έχουν εισαχθεί από τον contributor[3]
violations_eliminated	Ο συνολικός αριθμός violations που έχουν αφαιρεθεί από τον contributor

3. References

[1] “GitHub API” [Last accessed: 5 December 2018], Available at: https://developer.github.com/v3/

[2] Nagappan, N., Zeller, A., Zimmermann, T., Herzig, K., & Murphy, B., "*Change bursts as defect predictors*". In 2010 IEEE 21st International Symposium on Software Reliability Engineering (ISSRE), pp. 309-318.

[3] “PMD” [Last accessed: 5 December 2018], Available at: https://pmd.github.io/pmd-6.0.0/

[2] Για να χαρακτηριστεί ένας contributor μη ενεργός για μια μέρα θα πρέπει να μην έχει κάνει καμία αλλαγή στο repository.

[3] Ως violations ορίζονται παραβιάσεις ενδεικνυόμενων πρακτικών συγγραφής κώδικα [3]